

RANCANG BANGUN SISTEM STARTER SEPEDA MOTOR MENGUNAKAN SMARTPHONE BERBASIS MIKROKONTROLLER

Indranata U. Panggalo^{1*}, Rocky Yefrenes Dillak², Floriano J.X.S. Da Silva³

^{1,2,3}Politeknik Negeri Kupang

Jl. Adi Sucipto Penfui Kupang - NTT. P.O.Box. 139

*E-mail: indrapanggalo@gmail.com

rocky_dillak@yahoo.com

flo_dasilva@gmail.com

Abstrak

Meningkatnya penggunaan sepeda motor tidak terlepas dari tingkat kebutuhan, prestise, hobby maupun demografi sebuah daerah atau Negara. Indonesia merupakan salah satu negara berkembang dengan jumlah penduduk yang cukup padat yaitu mencapai 5 terpadat dunia. Dengan jumlah penduduk yang padat maka diikuti pula jumlah pengguna kendaraan bermotor. Saat ini menurut data BPS pertahun 2016, bahwa pengguna kendaraan bermotor 129 jutadan 105,15 juta adalah pengguna sepeda motor. Meningkatnya penggunaan sepeda motor ternyata diikuti pula oleh tingginya pencurian sepeda motor. Sistem keamanan pada sepeda motor, terutama pada bagian sistem kontak dan starter mesin. Sistem pengunci dari pabrikan belum mengalami perubahan berarti dan masih cenderung sama. Hal inilah yang menyebabkan pencuri leluasa beraktifitas membobol kendaraan roda dua yang memiliki nilai jual tinggi. Melalui penelitian ini, maka penulis memberikan sebuah alternative bagi sistem keamanan motor pada sisi sistem kontak dan starter motor. Dengan memanfaatkan alat/komponen tambahan yaitu *smartphone*, *bluetooth*, *mikrokontroller*, maka penulis berhasil membuat sebuah sistem *starter* pada sepeda motor dengan baik tanpa mengurangi/mempengaruhi kinerja dari kendaraan tersebut. Sistem kontak (*stand by*) pada motor hanya bisa diaktifkan menggunakan *smartphone* melalui *bluetooth*, untuk menghidupkan mesin secara elektrik (*dynamo starter*) juga hanya bisa dilakukan oleh *smartphone*, akan tetapi sistem *starter* manual (engkol) juga tetap bisa digunakan ketika kontak dalam keadaan "*stand by*".

Kata kunci: *Smartphone*, *Bluetooth*, *mikrokontroller*, *starter sepeda motor*

PENDAHULUAN

Seiring dengan perkembangan jaman, maka kebutuhan manusia akan alat transportasi semakin meningkat, salah satunya sepeda motor. Selain karena harganya yang relative terjangkau, bahan bakar yang irit, sepeda motor juga dianggap sebagai alat transportasi yang paling efektif dan efisien digunakan, terutama untuk jalan yang padat sehingga sedikit banyak bisa terhindar dari kemacetan. Hal inilah yang mendorong tingginya penggunaan sepeda motor di Indonesia. Berdasarkan data dari BPS bahwa pada tahun 2016, jumlah sepeda motor di Indonesia mencapai 105,15 juta unit atau lebih dari 81% jumlah kendaraan bermotor nasional. Seiring berkembangnya jumlah sepeda motor, tingkat pencurian sepeda motor juga semakin marak, terutama kepada sepeda motor yang memiliki nilai/harga jual tinggi.

Berbagai cara telah dilakukan oleh pemilik kendaraan sepeda motor, mengganti kunci standar pabrikan dengan menggunakan kunci khusus, menambah gembok pada cakram, maupun pengunci jari-jari dan sebagainya. Cara-cara tersebut tetap saja masih bisa "dibobol" oleh pencuri. Oleh karena itu melalui penelitian ini, penulis ingin menawarkan sebuah alternative baru bagi sistem keamanan motor khususnya pada sistem kontak da

starter motor menggunakan *handphone*.

Berbagai type *handphone* saat ini, salah satunya *smartphone*, menyediakan fitur-fitur yang bisa dimanfaatkan sebagai pengendali, apalagi didukung sistem operasi yang *open source*, memberikan kemudahan bagi kita untuk melakukan berbagai riset pengembangan sesuai dengan keinginan/kebutuhan. Salah satu fitur pada *smarthphone* yaitu Bluetooth dapat dimanfaatkan sebagai pengendali kepada bagi sistem kontak dan starter pada sepeda motor yang diproses oleh mikrokontroller.

Mikrokontroller merupakan salah satu pengendali berukuran mikro, yang dapat digunakan bersamaan dengan alat elektronika lainnya untuk aplikasi suatu sistem kendali. Pemakaian mikrokontroller umumnya digunakan dalam *embedded system*. Dengan kombinasi antara *smarthphone*, *bluetooth* dan *mikrokontroller* ini diharapkan akan menghasilkan sebuah sistem kontak dan pengendali *starter* menggantikan sistem dari pabrikan, dengan tujuan utama untuk sistem keamanan dari sepeda motor yaitu meminimalisir pencurian terhadap sepeda motor.

METODE PENELITIAN

Metode yang digunakan untuk merancang dan membangun sistem ini adalah dengan melakukan pendekatan secara model *waterfall* dimana sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematis, dengan beberapa tahapan, yaitu: *analysis, design, coding, testing* dan *maintenance*.

A. Analisis Kebutuhan Sistem

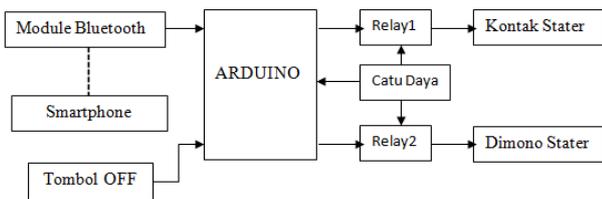
Langkah pertama dalam penelitian ini adalah, penulis melakukan observasi terhadap sistem kelistrikan dari sepeda motor, khususnya pada sistem instalasi kontak motor dan *starter* baik itu manual maupun elektrik. Tujuannya agar dapat dilakukan modifikasi sehingga sistem tersebut dapat dikendalikan oleh *smartphone* melalui *bluetooth* dan diproses oleh mikrokontroler.

Dari hasil observasi tersebut, penulis dapat menarik kesimpulan awal akan kebutuhan sistem yang diperlukan dan juga perangkat keras maupun lunak. Untuk kebutuhan *inputsistem* dibutuhkan sebuah *smarthphone* yang berbasis *android OS* dan memiliki *bluetooth* dan tombol *on/off*. Sedangkan perangkat keras lainnya utuk pengendali terdiri dari arduinouno, modul *bluetooth* dan *relay 12V*. Sedangkan perangkat lunak dibutuhkan pemograman *java* dengan menggunakan *IDEclipse* serta pemograman *C* dengan menggunakan *IdeArduino*.

B. Perancangan Sistem

1. Arsitektur Sistem

Arsitektur sistem yang dibangun untuk mendukung penelitian rancang bangun sistem *starter* sepeda motor menggunakan *smartphone* berbasis *mikrokontroler* yang dibuat adalah sebagai berikut :



Gambar 1. Arsitektur Sistem

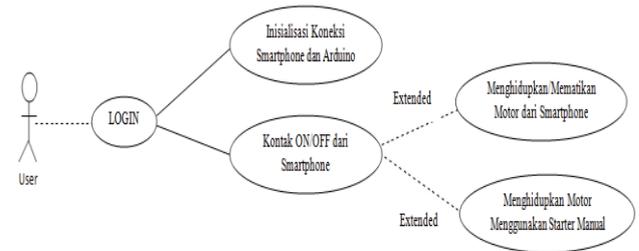
- Pada Gambar 1, terlihat bahwa rancang bangun *starter* sepeda motor menggunakan *smartphone* terdiri dari beberapa komponen, yaitu : *smartphone* sebagai pengontrol yang memberikan perintah untuk diproses oleh Arduino.
- Module *bluetooth* sebagai penerima data agar dapat diproses oleh arduino.
- *Relay* sebagai perangkat keluaran yang terhubung dengan kontak *starter* dan

dinamostarter, fungsinya untuk menghidupkan dan mematikan kontak *starter* dan *dinamostarter*.

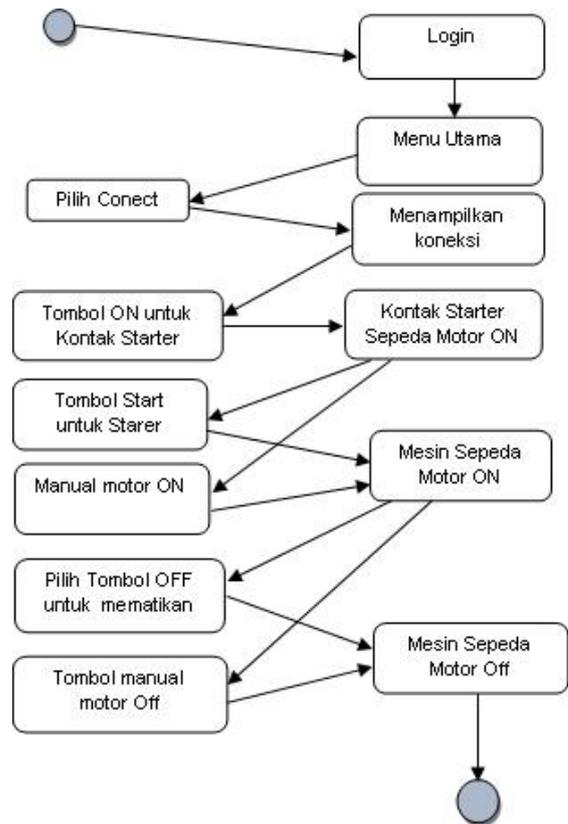
- Catu daya fungsinya memberikan daya ke arduino dan *relay*.
- Tombol fungsinya untuk mematikan *relay1* (kontak *starter*) secara manual.

2. Use Case dan activity Diagram Sistem

Deskripsi kerja sistem dapat dijabarkan menggunakan use case diagram seperti dibawah ini.



Gambar 2. Use Case Diagram Sistem



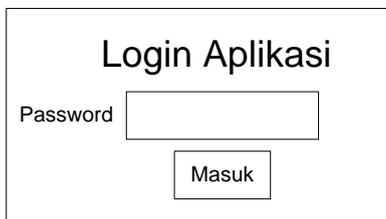
Gambar 3. Activity Diagram Sistem

Penjelasan *activity diagram*:

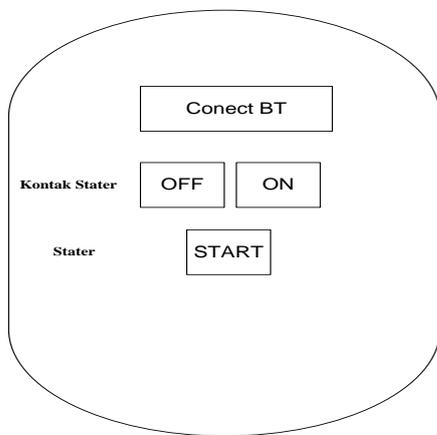
- Login, pada tahap ini *user* harus memasukkan *password* pada aplikasi untuk bisa menggunakan aplikasi.
- Inisialisasi koneksi *smartphone* dan arduino, pada tahap ini *user* memulai aplikasi dan melakukan koneksi dari *bluetoothsmartphone* ke module *bluetooth* pada *mikrokontroler*.

- Perintah ON atau OFF dari *smartphone*, tahap ini mengirimkan perintah dari *interface* di *smartphone* agar dapat diproses oleh *mikrokontroler*.
- Menghidupkan/mematikan motor dari *smartphone*, pada tahap ini sudah bisa menghidupkan atau mematikan mesin sepeda motor menggunakan *smartphone*.
- Menghidupkan motor menggunakan *starterengkol*, pada tahap ini mesin sepeda motor juga bisa dihidupkan dengan *starterengkol* atau secara manual.

3. Desain *Interface* aplikasi
 Pada *smartphone* akan dibuat sistem *interface* antara *user* dengan perangkat *smartphone*, untuk mengendalikan sistem.



Gambar 4. Tampilan Login



Gambar 5. Tampilan Main Menu

C. Pengkodean

Pada tahap pengkodean sistem, penulis menuliskan *coding* untuk membangun sistem terdiri dari 2 bahasa pemrograman :

- Pemograman *java*, menggunakan *IDE Eclipse*
- Pemograman *C*, menggunakan *Ide Arduino*.
- Menggunakan tools simulator (*proteus*) untuk melakukan pengujian *hardware* (*arduino* dan *relay*).

D. Pengujian Sistem

Pada tahap ini, penulis melakukan pengujian untuk mengetahui tingkat kebenaran dan akurasi

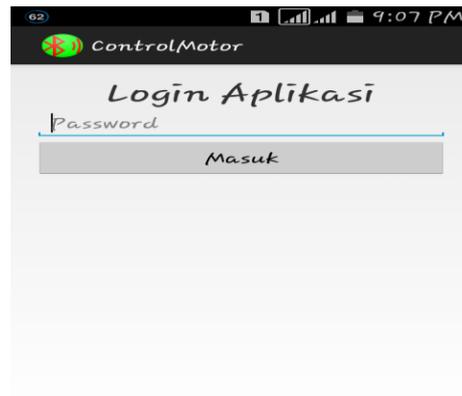
sistem. Proses-proses pengujian dilakukan pada *interfacehandphone*, koneksi *bluetooth*, pemrosesan *arduino* dan kinerja *relay*.

HASIL DAN PEMBAHASAN

HASIL

A. Aplikasi Android

Jalankan aplikasi yang telah dibuat pada *smartphone*, dan menginput *password* dan kemudian “masuk”.



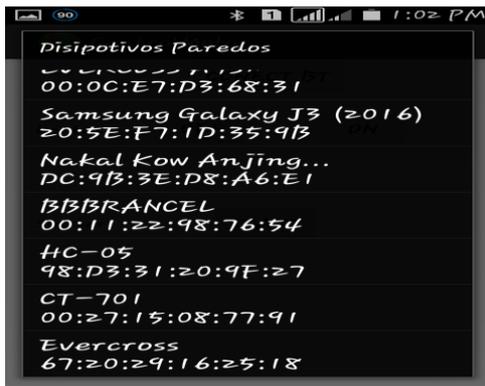
Gambar 6. Login Aplikasi

Setelah *user* berhasil *login*, akan muncul tampilan menu utama seperti gambar dibawah.



Gambar 7. Tampilan Menu Utama

Sebelum mengoperasikan aplikasi untuk dieksekusi oleh *mikrokontroler*, maka *user* harus melakukan koneksi ke modul *bluetooth* pada *arduino*, dengan tekan tombol “connect BT”. Setelah itu akan tampil daftar perangkat *bluetooth* yang terdeteksi dan aktif, kemudian *user* memilih HC-05.



Gambar 8. Perangkat Terhubung

Jika sudah terhubung maka tombol *ConectBT* akan berubah seperti pada gambar 9. Ini menandakan bahwa *smartphone* telah terkoneksi dengan modul *bluetooth* pada *arduino*. Sampai pada titik ini, *user* sudah bisa mengoperasikan perintah-perintah lainnya.

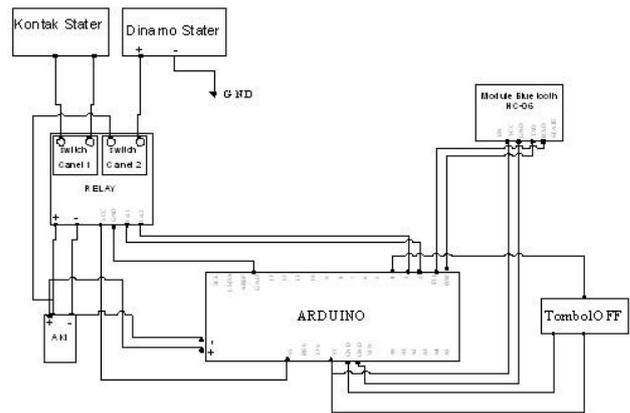


Gambar 9. Koneksi Antar Bluetooth

Setelah terkoneksi, maka *user* menekan tombol *ON* untuk menghidupkan kontak. Pada kondisi ini, mesin sepeda motor siap untuk dihidupkan. Mesin dapat dihidupkan secara elektrik dengan mendapat perintah dari *smartphone* (tekan *start*), atau dapat pula dihidupkan secara manual (engkol). Sedangkan tombol *OFF* untuk mematikan kontak sekaligus mesin sepeda motor.

B. Rangkaian Hardware

Gambar 10 merupakan gambar rangkaian hubungan antar perangkat keras dengan sistem kelistrikan motor khususnya untuk sistem *starter*.



Gambar 10. Rangkaian Hardware

Keterangan :

1. *Module Bluetooth* dan *Arduino*
 - Pin *VCC bluetooth* terhubung ke 5V pada *arduino*
 - Pin *GND bluetooth* terhubung ke *GND* pada *arduino*
 - Pin *TX bluetooth* terhubung ke Pin *RX* pada *arduino*
 - Pin *RX bluetooth* terhubung ke Pin *TX* pada *arduino*
2. *Relay* dan *Arduino*
 - Pin *VCC relay* terhubung ke Pin 5V pada *arduino*
 - Pin *GND relay* terhubung ke Pin *GND* pada *arduino*
 - Pin *int1 relay* terhubung ke Pin 2 pada *arduino*
 - Pin *int2 relay* terhubung ke Pin 3 pada *arduino*
3. *Tombol* dan *Arduino*
 - Pin + pada *tombol* terhubung ke Pin 5V dan Pin 4 pada *arduino*
 - Pin - pada *tombol* terhubung ke Pin *GND* pada *arduino*
4. *Relay* dan *Kontak Starter Switch* pada *relay* terhubung ke arus kiprok dan *coil* pada sepeda motor
5. *Relay* dan *Dinamo Starter Switch* pada *relay* terhubung ke *dinamo starter*
6. *AKI*
AKI memberikan daya 12V ke *arduino*, 12 V ke *relay* dan 12V ke *dinamo starter*.

C. Pengujian Sistem

Pengujian sistem telah dilakukan dan memberikan hasil sesuai dengan yang diharapkan. Pengujian dilakukan dengan beberapa metode, yaitu:

1. Pengujian Instruksi dari *smartphone* kepada sepeda motor.

Tabel 1. Pengujian Aplikasi pada *smartphone* dan sepeda motor.

No	Jenis Pengujian	Harapan	Hasil
1	Klik <i>icon</i> pada menu	Aplikasi running	Sesuai
2	Memasukkan <i>password</i> yang benar	Masuk ke menu utama	Sesuai
3	Memasukkan <i>assword</i> yang salah	Tetap di menu <i>login</i>	Sesuai
4	Menekan tombol <i>conectBT</i>	Tampil daftar perangkat <i>bluetooth</i> aktif	Sesuai
5	Memilih <i>bluetooth</i> HC-05	Terkoneksi dengan module <i>bluetooth</i>	Sesuai
6	Menekan tombol ON	Kontak hidup	Sesuai
7	Menekan tombol <i>start</i>	Motor hidup	Sesuai
8	<i>Starter</i> secara manua	motor hidup	Sesuai
9	Menekan tombol OFF	Motor mati	Sesuai
10	Menekan tombol secara manual	Motor mati	Sesuai

2. Pengujian Kompatibilitas
 Pengujian dilakukan pada beberapa *smartphone* android dengan versi yang berbeda dengan cara menginstal file *APK* kedalam *smartphone* dan selanjutnya mencoba semua fungsi tombol pada *interface*.

Tabel 2. Tabel Pengujian Komptibilitas Aplikasi Pada *Smartphone*.

No	Nama Perangkat	Versi	Jenis Pengujian			
			Pemasangan Aplikasi	Tampilkan Aplikasi	Tampilkan Daftar Bluetooth	Tampilkan Fungsi Tombol
1	Samsung J1 Ace	Sistem Operasi Android V5.1.1	Berhasil	Berhasil	Berhasil	Berhasil
2	Citricall CT-701	Lollipop Sistem Operasi Android v4.4 Kitkat	Berhasil	Berhasil	Berhasil	Berhasil
3	Oppo Neo 5	Sistem Operasi Android v6.0 Marshmallow	Berhasil	Berhasil	Berhasil	Berhasil

3. Pengujian kemampuan *Bluetooth* mengirim sinyal

Pengujian ini dilakukan dengan tujuan

mengetahui sejauh mana *Bluetooth* pada *smartphone* mengirim sinyal dan ditangkap oleh modul *bluetooth* pada sistem yang terpasang pada sepeda motor.

Tabel 3. Tabel Pengujian Kehandalan Koneksi Sesama *Bluetooth*

No	Jarak (Meter)	Hasil
1	1	Baik
2	5	Baik
3	10	Baik
4	15	Baik
5	20	Baik
6	23	Baik
7	24	Kurang baik
8	25	Buruk
9	26	Tidak ada koneksi

PEMBAHASAN

Pembuatan aplikasi system dibuat dalam beberapa tahap yaitu :

1. Pemograman Pada Arduino IDE

```
while (Serial.available() != 0) {
    perintah = perintah + char(Serial.read());
    Serial.println(perintah);
    delay(20) ;//tunda untuk baca perintah beriku
}
//kalau tidak ada data masukan, maka tidak per1
if (perintah.length() ==0)
perintah.trim(); // untuk proses kasi hilang te
{
    if (perintah == "TO")
    digitalWrite(RELAY_1, HIGH);
    else if (perintah == "TF")
    digitalWrite(RELAY_1, LOW);
    else if (perintah == "TS")
    digitalWrite(RELAY_2, HIGH);
    delay(1300);
    digitalWrite(RELAY_2, LOW);
    delay(10);
}
relayON = true;
{
    if (keadaanTegangan == HIGH)
    digitalWrite(RELAY_1, LOW);
}
}}
```

```
while (Serial.available() != 0) {
```

Fungsi dari perintah di atas adalah untuk membaca perintah dari serial jika perintah kosong maka arduino tidak memproses apa-apa.

```
perintah = perintah + char
(Serial.read());
Serial.println(perintah);
delay(20);
```

Fungsi dari perintah di atas adalah Serial akan membaca variabel perintah berupa karakter yang dikirim *smartphone*.

Sedangkan fungsi perintah untuk membaca karakter yang dikirim dari Serial berupa karakter "TO" untuk memberi perintah kontak starter "ON", perintah "TF" untuk memberi perintah kontak starter menjadi "OFF" dan perintah "TS" untuk menghidupkan motor secara elektrik.

2. Pembuatan *interface* android pada Eclipse IDE

a. Login

Berikut adalah pemrograman untuk tampilan login

```
<TextView
    android:id="@+id/data_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizo
    android:text="Login Aplikasi"
    android:textSize="30dp" />

<EditText
    android:layout_height="wrap_content"
    android:id="@+id/passwd_input"
    android:password="true" android:layout_wi
    android:hint="Password">
</EditText>

<Button
    android:id="@+id/masuk"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Masuk" >

</Button>
<TextView
    android:id="@+id/TextView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></Te
```

Perintah di atas untuk menampilkan elemen-elemen seperti *TextView*, *EditText* dan *Button* sesuai jarak yang ditentukan. Seperti perintah di bawah ini :

```
android:layout_height="wrap_content"
```

fungsi dari perintah di atas adalah lebar dari komponen *Widget* tersebut akan menyesuaikan content yang ada didalamnya. Misalnya *text* pada *button*. Lebar *button* akan menyesuaikan lebar *text* dalam *button* tersebut.

b. Halaman Utama

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.andro
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="3" >

    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="120dp"
        android:text="CONNECT_BT" />

    <Button
        android:id="@+id/off"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btn1"
        android:layout_marginLeft="32dp"
        android:layout_marginTop="25dp"
        android:layout_toRightOf="@+id/textView2"
        android:text="OFF" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/off"
```

Perintah di atas untuk menampilkan elemen-elemen seperti *TextView*, dan *Button* sesuai jarak yang ditentukan. Seperti perintah di bawah ini :

```
Android:layout_marginLeft="120dp"
```

Maksud dari perintah di atas adalah untuk menaruh elemen di sisi bagian kiri layar dengan jarak 120dp.

c. Pemrograman menu *login*

Berikut adalah pemrograman untuk menulogin

```
@Override
// method untuk override tombol masuk
public void onClick(View tombol_act) {
// TODO Auto-generated method stub
if (tombol_act == masuk) {

    if (password.getText().toString().equals(isi_passwd) ) {
        AlertDialog.Builder pesan = new AlertDialog.Builder(this);
        pesan.setMessage("Password anda benar")
        .setCancelable(false).setPositiveButton("OK",
        new DialogInterface.OnClickListener() {
```

Maksud dari perintah di atas adalah jika tombol masuk ditekan maka program akan membaca isi *password* dan jika *password*-nya benar maka akan tampil informasi bahwa *password* benar.

d. Pemrograman untuk koneksi *bluetooth*

Berikut adalah pemrograman untuk koneksi *bluetooth*

```
myBluetooth = BluetoothAdapter.getDefaultAdapter();

if (myBluetooth == null)
{
    Toast.makeText(getApplicationContext(), "Bluetooth Device Not Available", Toast.LENGTH_LONG).show();
}
else if (!myBluetooth.isEnabled())
{
    Intent turnBTon = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(turnBTon, TURN_ON);
}
}
```

Fungsi dari perintah di atas adalah jika *bluetoothdevice* tidak terpasang pada perangkat

maka program akan menampilkan informasi “Bluetooth Device Not Available”, dan jikaa *bluetooth device* terpasang paka program akan mengaktifkan *bluetooth* pada Perangkat.

e. Pemograman untuk kirim karakter ke arduino

```
private void turnOffled()
{
    if (mySocket!=null)
    {
        try
        {
            mySocket.getOutputStream().write("TF".toString().getBytes());
        }
        catch (IOException erro)
        {
            Toast.makeText(getApplicationContext(), " Terjadi erro :"+ erro, Toast.LENGTH_LONG).show()
        }
    }
}

private void turnStart()
{
    if (mySocket!=null)
    {
        try
        {
            mySocket.getOutputStream().write("TS".toString().getBytes());
        }
        catch (IOException erro)
        {
            Toast.makeText(getApplicationContext(), " Terjadi erro :"+ erro, Toast.LENGTH_LONG).show()
        }
    }
}
```

Penjelasan dari perintah di atas untuk mengirim karakter berupa TO, TF, dan TS ke Arduino melalui *OutputStream* ke *bluetoothsocket*.

PENUTUP

A. Kesimpulan

Berdasarkan hasil pegujian terhadap sistem yang telah dibuat, maka penulis dapat menarik beberapa kesimpulan :

1. Pembuatan sistem *starter* menggunakan *smartphone* dan *mikrokontroler* berhasil dibuat dengan media komunikasi *bluetooth*.
2. Pembuatan aplikasi *interface* android sebagai pengontrol dari *smartphone* berhasil dibuat dan dapat berjalan dengan baik pada android OS versi 4.4 keatas.
3. Pengujian koneksi antara *Bluetooth* pada *smartphone* dengan modul *bluetooth* yang terhubung pada *arduino* dapat berjalan dengan baik dan jarak maksimal koneksi adalah 23 meter.
4. Sistem kontak motor dapat dikendalikan secara penuh menggunakan *smartphone*, dan motor dapat distarter maupun dimatikan menggunakan aplikasi android.
5. Sistem *starter* secara manual (engkol) tetap dapat berfungsi dengan baik, setelah sistem kontak dihidupkan oleh *smartphone*.

B. Saran

Tujuan utama dari membangun sistem ini adalah member keamanan bagi pengguna motor agar dapat meminimalisir terhadap pecuria kendaraan bermotor, oleh karea itu penulis merasa

belumlah tentu sistem yang kami buat ini dapat menjawab secara penuh terhadap kebutuhan *user*, oleh karena itu penulis memiliki beberapa saran terkait pengembangan sistem ini, yaitu :

1. Sistem dapat ditambahkan sistem *GPS Tracker*, sehingga dapat dikontrol pergerakan sepeda motor berdasarkan titik koordinat.
2. Sistem yang dibangun ini meggunakan *arduino uno*, tetapi dapat juga menggunakan *arduino nano* karena secara fisik lebih kecil dan harganya lebih murah.

UCAPAN TERIMAKASIH

Penelitian rancang bangun sistem *starter* pada sepeda motor berbasis *smartphone* dan *mikrokontroler* ini tidak akan selesai tanpa bantuan dari rekan-rekan peneliti, oleh karena itu penulis mengucapkan terima kasih kepada rekan-rekan peneliti, karena kita kita telah dapat bekerja sama dengan baik dan mampu menghasilkan prototype ini. Kiranya kebersamaan kita dalam penelitian ini tetap dapat dipertahankan untuk penelitian selanjutnya maupun pengembangan prototype yang dihasilkan saat ini.

DAFTAR PUSTAKA

Andrianto H. 2008. *Pemrograman Mikrokontroler AVR ATmega8 menggunakan bahasa C*. Bandung: Informatika.

Artanto D. 2012. *Interaksi Arduino dan LabVIEW*. Jakarta: Elex Media Komputindo.

Harinoto, Fendi. 2007. *Sistem Pengaman Sepeda Motor Via RFID dan sms menggunakan mikrokontroler atmega 128*. Jember : Universitas Muhammadiyah.

Nazarudin S. 2010. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.

Putu, Giovanni. 2012. *Mengenal Relay*. Jakarta : Penerbit Informatika.

Dillak, R. (2017). KLASIFIKASI MUSIK MENGGUNAKAN POLYNOMIAL NEURAL NETWORK. *Jurnal Ilmiah Flash*, 3(2), 94-99

Safaat H, & Nazruddin.2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Penerbit Informatika.

Tjhin, Santo, 2014, *Sistem Keamanan Sepeda Motor Melalui Short Message Service Menggunakan AVR Mikrokontroler atmega8*. *Skripsi Terpublikasi*, Yogyakarta: STMIK Raharja.

Winoto A. 2010. *“Mikrokontroler AVR ATmega8/16/32/8535 dan Pemrograman dengan Bahasa C pada WinAVR*, Bandung: Penerbit Informatika Bandung.